



COPY OF PAPERS
ORIGINALLY FILED

Docket No. 2001-068P1
PATENT

CERTIFICATE OF MAILING

I hereby certify that this correspondence is being deposited with the United States Postal Service with sufficient postage as First Class Mail in an envelope addressed to: Assistant Commissioner for Patents, Washington, D.C. 20231 on 5/25/02.

BY:

Suzanne Shadley

Suzanne Shadley

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:	Benoit et al.
Serial No.:	60/361,467
Filed:	03/04/02
For:	Copolymers For Use In Capillary Electrophoretic Separation Media

Group Art Unit: Not Assigned

Examiner: Not Assigned

Assistant Commissioner for Patents
Washington D.C. 20231

RECEIVED

MAY 13 2002

Technology Center 2100

TRANSMITTAL LETTER

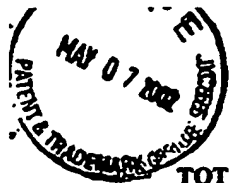
Sir:

Transmitted herewith (check all that apply):

- | | |
|--|--|
| <input type="checkbox"/> Preliminary Amendment | <input type="checkbox"/> Information Disclosure Statement |
| <input type="checkbox"/> Response/Amendment | <input type="checkbox"/> Petition Under 37 CFR 1.97(d)(2) |
| <input type="checkbox"/> Response/Amendment After Final | <input type="checkbox"/> Formal Drawings |
| <input type="checkbox"/> Supplemental Amendment | <input type="checkbox"/> Declaration Under 37 CFR 1.131 |
| <input type="checkbox"/> Affidavits/Declarations | <input type="checkbox"/> Declaration Under 37 CFR 1.132 |
| <input type="checkbox"/> Declaration and Power of Attorney | <input type="checkbox"/> Terminal Disclaimer |
| <input type="checkbox"/> Supplemental Declaration | <input type="checkbox"/> Small Entity Statement |
| <input type="checkbox"/> Power of Attorney | <input type="checkbox"/> Request for Refund |
| <input type="checkbox"/> Change of Correspondence Address | <input type="checkbox"/> Appeal |
| <input type="checkbox"/> Associate Power of Attorney | <input checked="" type="checkbox"/> Petition to Correct Inventorship Under |
| <input checked="" type="checkbox"/> Amended Application Data Sheet | 37 CFR §1.48(d) |

to be filed in the above-identified patent application.

This Page Blank (uspto)



COPY OF PAPERS
ORIGINALLY FILED

TOTAL AMOUNT OF PAYMENT: \$50.00

METHOD OF PAYMENT

The Commissioner is hereby authorized to charge indicated fees and credit any overpayments to:

Deposit Account Number: 50-0496

Deposit Account Name: Symyx Technologies, Inc.

☒ Charge Any Additional Fee Required Under 37 CFR 1.16 and 1.17

☐ Charge the Issue Fee Set at 37 CFR 1.18 at the Mailing of the Notice of Allowance

RECEIVED

MAY 13 2002

Technology Center 2100

FEE CALCULATION

1. EXTRA CLAIM FEES

	Extra Claims		Fee from below		Fee Paid	
Total Claims	-20**	=	X		=	
Independent Claims	-3**	=	X		=	
Multiple Dependent Claims (first appearance) \$280/140						

**or number previously paid, if greater; For Reissues, see below

Large Fee Code	Entity Fee (\$)	Small Fee Code	Entity Fee (\$)	Fee Description
103	18	203	9	Claims in excess of 20
102	84	202	42	Independent claims in excess of 3
104	280	204	140	Multiple dependent claim, if not paid
109	84	209	42	**Reissue independent claims over original patent
110	18	210	9	**Reissue claims in excess of 20 and over original patent
				SUBTOTAL (2) (\$)

This Page Blank (uspto)



COPY OF PAPERS
ORIGINALLY FILED

Docket No. 2001-068P1
PATENT

2. ADDITIONAL FEES

Large Fee Code	Entity Fee (\$)	Small Fee Code	Entity Fee (\$)	Fee Description	Fee Paid
105	130	205	65	Surcharge - late filing fee or oath	RECEIVED MAY 13 2002 Technology Center 2100
127	50	227	25	Surcharge - late provisional filing fee or cover sheet	
139	130	139	130	Non-English specification	
147	2,520	147	2,520	For filing a request for reexamination	
112	920*	112	920*	Requesting publication of SIR prior to Examiner action	
113	1,840*	113	1,840*	Requesting publication of SIR after Examiner action	
115	110	215	55	Extension for reply within first month	
116	400	216	200	Extension for reply within second month	
117	920	217	460	Extension for reply within third month	
118	1440	218	720	Extension for reply within fourth month	
128	1,960	228	980	Extension for reply within fifth month	
119	320	219	160	Notice of Appeal	
120	320	220	160	Filing a brief in support of an appeal	
121	280	221	140	Request for oral hearing	
138	1,510	138	1,510	Petition to institute a public use proceeding	
140	110	240	55	Petition to revive - unavoidable	
141	1,280	241	640	Petition to revive - unintentional	
142	1,280	242	640	Utility issue fee (or reissue)	
143	460	243	230	Design issue fee	
144	620	244	310	Plant issue fee	
122	130	122	130	Petitions to the Commissioner	
123	50	123	50	Petitions related to provisional applications	
126	180	126	180	Submission of Information Disclosure Statement	
581	40	581	40	Recording each patent assignment per property (times number of properties)	
146	740	246	370	Filing a submission after final rejection (37 CFR 1.129(a))	
149	740	249	370	For each additional invention to be examined (37 CFR 1.129(b))	
Other fee (specify) Petition to correct inventorship 37 CFR §1.17(g)					\$50.00
SUBTOTAL (3) (\$)					50.00

*Reduced by Basic Filing Fee Paid

Respectfully submitted,

Ronald A. Krasnow
Reg. No. 33,321
Attorney for Applicant(s)

Date: 4-28-02

Symyx Technologies
3100 Central Expressway
Santa Clara, California 95051
Tel.: (408) 764-2000 Fax: (408) 773-4029

This Page Blank (uspiv,



COPY OF PAPERS
ORIGINALLY FILED

6P/2151

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant(s): Thorsten Laux

Assignee: Sun Microsystems, Inc.

Title: A SYSTEM AND METHOD INCLUDING A MERGING DRIVER
FOR ACCESSING MULTIPLE DATA SOURCES

Serial No.: 09/738,464 Filed: December 13, 2000

Examiner: Unknown Group Art Unit: 2151

Docket No.: P-4589

Monterey, CA
April 26, 2002

ASSISTANT COMMISSIONER FOR PATENTS
WASHINGTON, DC 20231

RECEIVED

MAY 13 2002

CLAIM FOR PRIORITY AND
SUBMISSION OF PRIORITY DOCUMENT

Technology Center 2100

Dear Sir:

Enclosed is a certified copy of the foreign priority application no. 99124939.2 for the above application.

Applicant claims the foreign priority filing date of December 14, 1999 on which the enclosed foreign priority application was filed.

CERTIFICATE OF MAILING

I hereby certify that this correspondence is being deposited with the United States Postal Service with sufficient postage as First Class Mail in an envelope addressed to: Assistant Commissioner for Patents, Washington, DC 20231, on April 26, 2002.

Respectfully submitted,

Forrest Gunnison
Attorney for Applicant(s)
Reg. No. 32,899
(831) 655-0880

Attorney for Applicant(s) Date of Signature April 26, 2002

GUNNISON, McRAY &
RODGSON, L.L.P.
One West Office Plaza, Suite 220
1900 Gordon Road
Monterey, CA 93940
Tel. 831-453-0880
Fax 831-453-0888

This Page Blank (uspio,



Europäisches
Patentamt

European
Patent Office

Office européen
des brevets

09/738464

Bescheinigung

Certificate

Attestation

Die angehefteten Unterlagen stimmen mit der ursprünglich eingereichten Fassung der auf dem nächsten Blatt bezeichneten europäischen Patentanmeldung überein.

The attached documents are exact copies of the European patent application described on the following page, as originally filed.

Les documents fixés à cette attestation sont conformes à la version initialement déposée de la demande de brevet européen spécifiée à la page suivante.

Patentanmeldung Nr. Patent application No. Demande de brevet n°

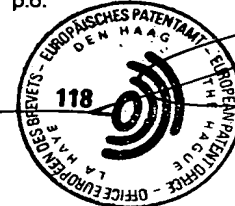
99124939.2

**CERTIFIED COPY OF
PRIORITY DOCUMENT**

Der Präsident des Europäischen Patentamts;
Im Auftrag

For the President of the European Patent Office

Le Président de l'Office européen des brevets
p.o.



M. de Roo

MÜNCHEN, DEN
MUNICH,
MUNICH, LE

22/03/02

This Page Blank (uspto)

0900 7786

(2) ϕ has a left inverse ψ if and only if ϕ is injective and ψ is a left inverse of ϕ .

100-443887-100

1925

the 1990s, the number of people in the world who are under 15 years of age is expected to increase from 1.1 billion to 1.5 billion. The number of people aged 65 and over is expected to increase from 250 million to 450 million. The number of people aged 15 and over is expected to increase from 3.5 billion to 4.5 billion. The number of people aged 15 and over is expected to increase from 3.5 billion to 4.5 billion. The number of people aged 15 and over is expected to increase from 3.5 billion to 4.5 billion.

1977-1978 1979-1980 1981-1982 1983-1984 1985-1986 1987-1988 1989-1990 1991-1992 1993-1994 1995-1996 1997-1998 1999-2000 2001-2002 2003-2004 2005-2006 2007-2008 2009-2010 2011-2012 2013-2014 2015-2016 2017-2018 2019-2020 2021-2022 2023-2024 2025-2026 2027-2028 2029-2030 2031-2032 2033-2034 2035-2036 2037-2038 2039-2040 2041-2042 2043-2044 2045-2046 2047-2048 2049-2050 2051-2052 2053-2054 2055-2056 2057-2058 2059-2060 2061-2062 2063-2064 2065-2066 2067-2068 2069-2070 2071-2072 2073-2074 2075-2076 2077-2078 2079-2080 2081-2082 2083-2084 2085-2086 2087-2088 2089-2090 2091-2092 2093-2094 2095-2096 2097-2098 2099-2100 2101-2102 2103-2104 2105-2106 2107-2108 2109-2110 2111-2112 2113-2114 2115-2116 2117-2118 2119-2120 2121-2122 2123-2124 2125-2126 2127-2128 2129-2130 2131-2132 2133-2134 2135-2136 2137-2138 2139-2140 2141-2142 2143-2144 2145-2146 2147-2148 2149-2150 2151-2152 2153-2154 2155-2156 2157-2158 2159-2160 2161-2162 2163-2164 2165-2166 2167-2168 2169-2170 2171-2172 2173-2174 2175-2176 2177-2178 2179-2180 2181-2182 2183-2184 2185-2186 2187-2188 2189-2190 2191-2192 2193-2194 2195-2196 2197-2198 2199-2200 2201-2202 2203-2204 2205-2206 2207-2208 2209-2210 2211-2212 2213-2214 2215-2216 2217-2218 2219-2220 2221-2222 2223-2224 2225-2226 2227-2228 2229-2230 2231-2232 2233-2234 2235-2236 2237-2238 2239-2240 2241-2242 2243-2244 2245-2246 2247-2248 2249-2250 2251-2252 2253-2254 2255-2256 2257-2258 2259-2260 2261-2262 2263-2264 2265-2266 2267-2268 2269-2270 2271-2272 2273-2274 2275-2276 2277-2278 2279-2280 2281-2282 2283-2284 2285-2286 2287-2288 2289-2290 2291-2292 2293-2294 2295-2296 2297-2298 2299-2300 2301-2302 2303-2304 2305-2306 2307-2308 2309-2310 2311-2312 2313-2314 2315-2316 2317-2318 2319-2320 2321-2322 2323-2324 2325-2326 2327-2328 2329-2330 2331-2332 2333-2334 2335-2336 2337-2338 2339-2340 2341-2342 2343-2344 2345-2346 2347-2348 2349-2350 2351-2352 2353-2354 2355-2356 2357-2358 2359-2360 2361-2362 2363-2364 2365-2366 2367-2368 2369-2370 2371-2372 2373-2374 2375-2376 2377-2378 2379-2380 2381-2382 2383-2384 2385-2386 2387-2388 2389-2390 2391-2392 2393-2394 2395-2396 2397-2398 2399-2400 2401-2402 2403-2404 2405-2406 2407-2408 2409-2410 2411-2412 2413-2414 2415-2416 2417-2418 2419-2420 2421-2422 2423-2424 2425-2426 2427-2428 2429-2430 2431-2432 2433-2434 2435-2436 2437-2438 2439-2440 2441-2442 2443-2444 2445-2446 2447-2448 2449-2450 2451-2452 2453-2454 2455-2456 2457-2458 2459-2460 2461-2462 2463-2464 2465-2466 2467-2468 2469-2470 2471-2472 2473-2474 2475-2476 2477-2478 2479-2480 2481-2482 2483-2484 2485-2486 2487-2488 2489-2490 2491-2492 2493-2494 2495-2496 2497-2498 2499-2500 2501-2502 2503-2504 2505-2506 2507-2508 2509-2510 2511-2512 2513-2514 2515-2516 2517-2518 2519-2520 2521-2522 2523-2524 2525-2526 2527-2528 2529-2530 2531-2532 2533-2534 2535-2536 2537-2538 2539-2540 2541-2542 2543-2544 2545-2546 2547-2548 2549-2550 2551-2552 2553-2554 2555-2556 2557-2558 2559-2560 2561-2562 2563-2564 2565-2566 2567-2568 2569-2570 2571-2572 2573-2574 2575-2576 2577-2578 2579-2580 2581-2582 2583-2584 2585-2586 2587-2588 2589-2590 2591-2592 2593-2594 2595-2596 2597-2598 2599-2600 2601-2602 2603-2604 2605-2606 2607-2608 2609-2610 2611-2612 2613-2614 2615-2616 2617-2618 2619-2620 2621-2622 2623-2624 2625-2626 2627-2628 2629-2630 2631-2632 2633-2634 2635-2636 2637-2638 2639-2640 2641-2642 2643-2644 2645-2646 2647-2648 2649-2650 2651-2652 2653-2654 2655-2656 2657-2658 2659-2660 2661-2662 2663-2664 2665-2666 2667-2668 2669-2670 2671-2672 2673-2674 2675-2676 2677-2678 2679-2680 2681-2682 2683-2684 2685-2686 2687-2688 2689-2690 2691-2692 2693-2694 2695-2696 2697-2698 2699-2700 2701-2702 2703-2704 2705-2706 2707-2708 2709-2710 2711-2712 2713-2714 2715-2716 2717-2718 2719-2720 2721-2722 2723-2724 2725-2726 2727-2728 2729-2730 2731-2732 2733-2734 2735-2736 2737-2738 2739-2740 2741-2742 2743-2744 2745-2746 2747-2748 2749-2750 2751-2752 2753-2754 2755-2756 2757-2758 2759-2760 2761-2762 2763-2764 2765-2766 2767-2768 2769-2770 2771-2772 2773-2774 2775-2776 2777-2778 2779-2780 2781-2782 2783-2784 2785-2786 2787-2788 2789-2790 2791-2792 2793-2794 2795

Journal of Management Studies, 37(6), 809-826.



Europäisches
Patentamt

European
Patent Office

Office européen
des brevets

Blatt 2 der Bescheinigung
Sheet 2 of the certificate
Page 2 de l'attestation

Anmeldung Nr.:
Application no.: 99124939.2
Demande n°:

Anmeldetag:
Date of filing: 14/12/99
Date de dépôt:

Anmelder:
Applicant(s):
Demandeur(s):
SUN MICROSYSTEMS, INC.
Palo Alto, California 94303
UNITED STATES OF AMERICA

Bezeichnung der Erfindung:
Title of the invention:
Titre de l'invention:

Merging driver for accessing multiple database sources

In Anspruch genommene Priorität(en) / Priority(ies) claimed / Priorité(s) revendiquée(s)

Staat:
State:
Pays:

Tag:
Date:
Date:

Aktenzeichen:
File no.
Numéro de dépôt:

Internationale Patentklassifikation:
International Patent classification:
Classification internationale des brevets:

G06F17/30

Am Anmeldetag benannte Vertragsstaaten:
Contracting states designated at date of filing:
Etats contractants désignés lors du dépôt:

AT/BE/CH/CY/DE/DK/ES/FI/FR/GB/GR/IE/IT/LI/LU/MC/NL/PT/SE

Bemerkungen:
Remarks:
Remarques:

UNITED STATES DEPARTMENT OF COMMERCE

OFFICE OF THE SECRETARY

WASHINGTON, D. C. 20540

FOR THE SECRETARY OF COMMERCE

ADMINISTRATIVE INFORMATION

1. The purpose of this document is to provide information regarding the procedures for the submission of applications for the registration of trademarks and service marks. This document is intended for the use of applicants and is not to be construed as a legal opinion. The information contained herein is subject to change without notice.

This Page Blank (uspto)

2. The information contained in this document is for informational purposes only and is not to be construed as a legal opinion. The information contained herein is subject to change without notice. The information contained herein is for informational purposes only and is not to be construed as a legal opinion. The information contained herein is subject to change without notice.

UNITED STATES DEPARTMENT OF COMMERCE

OFFICE OF THE SECRETARY
WASHINGTON, D. C. 20540
FOR THE SECRETARY OF COMMERCE

14. Dez. 1999

MERGING DRIVER FOR ACCESSING MULTIPLE DATABASE SOURCES

FIELD OF THE INVENTION

5 The present invention relates to the access of a database having multiple database sources, and more particularly, the invention relates to the access of said multiple database sources by a single search or query.

DESCRIPTION OF THE RELATED ART

10

 There are known databases where the data to be accessed is distributed over multiple data sources, such as different directories, different servers, etc., and where the different data sources have different formats. In this case in known systems only one data source can be accessed by a single search or a query. To
15 access multiple sources in known systems the user has to dispatch multiple queries directed to the multiple sources.

 There is therefore a need for providing the user with the capability to search by a single query data distributed over more than one data source, since the
20 data needed by the user may be spread over the different sources containing respectively different data tables or databases of different formats.

SUMMARY OF THE INVENTION

25

 According to an aspect of the present invention there is provided a method for accessing multiple database sources of a database by a single access or query. The multiple sources may have different formats and may need to be accessed by different drivers, and there is provided a common application
30 programming interface (API) for individually accessing the multiple sources through said corresponding drivers. The access to multiple sources by a single access operation from the user can be achieved by the provision of a merging driver which

accesses the plurality of sources by accessing the corresponding drivers through a common API.

In other words, for each source of a different format there is a
5 corresponding driver, those plurality of different drivers can be accessed by a common API, and a merging driver utilizes the common API for attaining a merging access to a plurality of sources requiring different drivers, by causing the common API to perform access operations to said plurality of sources in response to a single
10 access to the merging driver. The merging driver therefore acts as a kind of a generator for generating a plurality of queries to different sources having different formats or requiring different drivers in response to a single query to the merging driver.

It may therefore be said that the provision of the merging driver provides
15 a kind of a "virtual" database, where multiple sources requiring individually different drivers are accessed under the roof of a common API, the merging driver using the common API to access the multiple sources such that they can be queried through a single query. An application program or a user interface may access the merging driver through its API to access multiple sources by a single access operation.

20 Preferably the merging driver can be configured by a user in order to define which of the available sources are to be simultaneously accessed through a single access to the merging driver.

25 Preferably from a user's point of view the merging driver is configured such that it looks to the outside world, or better to say to the user interface, like a database source itself. This means that the user can access the merging driver like a single source, however, this single source actually comprises multiple database sources which are accessed by a single access operation.

30 Preferably there is provided an option to perform an autocompletion procedure for the outputs obtained through the single access achieved by said

merging driver to multiple sources. It is then for example possible that a user configures a merging driver for those sources only for which performing an autocompletion makes sense, which means for sources which are not too large to perform an autocompletion procedure, since an autocompletion requires a lot of RAM and processing power.

The multiple sources can be databases which are separate from one another, they may be in different locations, may be of different kinds, and may require the use of different query engines to access their data. For example, one may be a StarBase database, another may be a database of another kind accessible via a corporate intranet or via the internet, etc. The user can through the merging driver access such multiple databases with a single query and work with the data collected from these databases as if they had been obtained from a single database. Thereby there is created a kind of "virtual database" based on the multiple sources (which may themselves be different databases).

A user can specify the sources or databases of the "virtual database" to be accessed, and defines a query or any other access operation to all of the sources specified. The user's query is translated by the merging driver into corresponding queries which are addressed to the specified sources (or databases), and then retrieves the requested data from these databases. The so retrieved data may then be used to generate a serial letter, a form letter, or the like by an application program such as a word processor.

The user can therefore retrieve data from multiple and dissimilar databases by using a single user interface and a single query language, and the retrieved data can be manipulated as a whole rather in segments according to source, so that operations such as sorting are carried out on all of the retrieved data.

The great advantage and utility of the present invention lies in the fact that a user by a single query (or any other access operation, e.g. for editing) can

access data which may be distributed over multiple sources, let them be databases, database tables, or any other sources which may require different drivers for being accessed due to their different formats. By specifying the sources to be accessed a user can "merge" different sources into a "virtual" database which looks to the user like a single database but which is actually a grouping of several databases or sources selected by the user, so that a single query can provide him with results which are retrieved from different sources.

If the database is for example an address book, then the present invention enables the user to generate serial letters based on a single address book query, although the addresses he needs may be distributed over multiple sources which themselves are accessed through different drivers.

Further features and advantages will become apparent from the following detailed description which explains the present invention through preferred embodiments in connection with the accompanying drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 shows a block diagram of a computer system to be used with a preferred embodiment of the present invention.

Fig. 2A schematically shows a database having different data sources.

Fig. 2B schematically shows a database having different data sources, one of them being a merging source.

Fig. 3 schematically shows an access to a database having different data sources.

Fig. 4 schematically shows the access to multiple data sources by a single query according to a preferred embodiment of the present invention.

Fig. 5A schematically shows the access of the multiple sources through a single query.

5 Fig. 5B schematically shows the obtaining and ordering of the results.

Fig. 6 schematically shows a user's perspective of an access as illustrated in Fig. 4.

10 Fig. 7 shows a flowchart of a database access according to a preferred embodiment of the present invention.

Fig. 8 schematically illustrates a merging driver for accessing multiple sources.

15 Fig. 9 shows the configuration of a merging driver according to a preferred embodiment of the present invention.

20 Fig. 10 shows a flowchart illustrating in more detail the accessing step shown in Fig. 7.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

Fig. 1 schematically shows the configuration of a computer system to be used in connection with the preferred embodiment of the present invention. The computer 100 contains a CPU 110, a memory 120, and an I/O-unit 130. The computer 100 is capable of executing programs by carrying out computer instructions through CPU 110 which the CPU fetched from memory 120 and which may have been stored in a storage device 150 such as a CD-ROM or a floppy disk.

25 The I/O-unit 130 is connected to a keyboard 160 and a mouse 170 to enable a user to input data to the computer, and it is further connected to a printer 180 to output

documents as hardcopies. Computer 100 is further connected to a display unit 140 such as a monitor.

Those skilled in the art will readily recognize that the foregoing combination is just an illustrative example and not to be understood as limiting the features and advantages of the present invention as outlined in the following.

Fig. 2A shows an example for a database which has multiple sources containing data of different formats and requiring different drivers for being accessed. The example shown in Fig. 2 is the address book of the StarOffice Software Package produced by Sun Microsystems, Palo Alto, CA. Fig. 2A shows the address book in a tree view as seen by the user when the StarOffice software is running on his computer.

The node 200 of the tree is labelled address book, and the leaves of the tree are the individual sources SOURCE1 210, SOURCE2 220, SOURCE3 230, and SOURCE4 240. The SOURCE1 210 is now for example a LDAP source, which means that the data in this source is stored according to the LDAP (lightweight directory access protocol) format and have to be accessed by a LDAP driver. The source SOURCE2 220 may be a database where data is stored in tables according to a certain database software, such as StarBase being part of the StarOffice package, produced by Sun Microsystems, Palo Alto, CA. This means that the SOURCE2 220 is to be accessed by a different driver, which in fact is the database software StarBase.

The source SOURCE3 230 may be another source also requiring a certain driver different from the ones for the source 210 and source 220. Finally, the source "SOURCE4" 240 is another source requiring again a different driver, it may for example be the source "SCHEDULE" containing the users which are registered as users of the StarOffice schedule software, being part of the StarOffice Suite produced by Sun Microsystems, Palo Alto, CA.

Each of the sources 210 to 240 as well as the corresponding node 200 are to be understood as being for exemplary purposes only, the present invention may be employed in connection with any grouping of different sources which may themselves be databases of different kinds and which can be accessed through a common front end shown as element 200 in Fig. 2A. It will later be described how a single access through element 200 can retrieve data from a plurality of the sources shown in Fig. 2A.

The grouping of several sources 210 to 240 requiring different drivers and their access through the common front end or user interface 200 shown in Fig. 2A will now be further explained in connection with Fig. 3. Fig. 3 shows the access to different sources 320, 330, and 340 by a user through user interface 300. Each of the different sources 320, 330, and 340 is accessed by a correspondingly different driver 315, 325, and 335. From the side of a user interface 300 these different drivers can be accessed through a common API 310. This means that a user can access through user interface 300 the API 310 which is the same for all of the different drivers 315, 325, and 335. For the user interface 300 there are different sources which may be accessed, however, there is only one single API through which such an access to the different sources can be carried out. Based on such a configuration and by further providing a merging driver described in the following, it becomes possible to simultaneously access a plurality of different sources by a single access operation. This will now be described in more detail in the following.

Fig. 4 schematically illustrates how by a single access operation (such as a query) multiple sources 430, 440, and 450 can be accessed as if they were a single source (a single database). For that purpose there is provided the merging driver 410 which has the task of distributing a query (or any other access operation) which it has received from user interface 400 through its own API 405 to the different drivers 425, 435, and 445 through their corresponding API 420, which is identical for all of those different drivers. The merging driver 410 therefore performs a kind of a demultiplexing operation for any access operation which it receives. Any

corresponding instruction or sequence of instructions is just distributed through the different sources through their corresponding drivers and their common API.

Fig. 5A schematically shows a flow chart illustrating the operation of the merging driver. The merging driver in step 510 receives a query (or any other access instruction) from a user interface, it distributes the query to the different drivers in step 520, in step 530 it obtains and orders the results from the drivers, for example alphabetically or according to any other criterion, and finally in step 540 the result is output to the user interface.

Ordering the result can be performed together with obtaining the result by obtaining from a source only a result if it matches the ordering criterion. If the ordering criterion is "order alphabetically according to title", then for example a result starting with character "B" from one source is not obtained as long as the other sources do not offer higher ranking results, such as results starting with character "A".

This is schematically illustrated in Fig. 5B. An result offered by a source (or its driver) in step 500 is checked in step 510 as to whether there exist higher-ranking results based on the ordering criterion. Examples for an ordering criterion are "order alphabetically" or "order numerically". If another source offers such a higher-ranking result then in step 520 attention is turned to this source, then returning to step 500. If no higher-ranking result is offered, then the offered result is obtained in step 530. The result may then be returned as an output, it may, however, also be buffered to return a plurality of outputs which have been buffered simultaneously. Step 540 checks whether there are more results, if so then it returns to step 500. If no further results are offered, then the obtaining and ordering of the results is finished in step 550. In accordance with the foregoing the result may be obtained in an ordered manner.

A particular feature of the configuration explained hereinabove, namely how a merging driver looks like from the side of the user interface, is explained in

connection with Fig. 6. Fig. 6 shows a user interface 600, sources of different kinds 620, 630, 640, and 650 (the corresponding drivers here are not shown) and an API 610 identical for all of the different sources. In Fig. 6, however, source 650 may be a "merging source", which means that with an access to source 650 actually a plurality of different databases which are dissimilar is accessed at one time. However, since those different databases all are accessed through the same API, and since the merging driver explained before distributes a query to the different sources, from the outside world such a grouping of sources accessed through a merging driver does not look different than an individual source itself. Such a grouping of different sources (or different databases) accessed through the merging driver from the outside world looks like any other source which can be accessed through the single API 610, although it actually consists of multiple sources such as sources 660, 670, and 680 shown in Fig. 6. Those multiple sources are accessed through their corresponding merging driver 690, and since this merging driver makes use of the common API of the sources 660, 670, and 680 (which is identical to API 610 of the other sources) it can itself be accessed through API 610. In other words, the merging driver is implemented such that from the outside world it can be accessed through the same API 610 as the other sources 620 to 640.

Those skilled in the art will readily recognize that with such a concept it also becomes possible that one of the sources 660 to 680 which are merged by merging driver 690 as explained in connection with Fig. 6 may itself be a source consisting of different sources (a "merging source") as element 650 in Fig. 6.

Therefore, through the concept of the provision of a common API for all of the different drivers, and further through the provision of a merging driver performing a kind of a demultiplex operation distributing a query to the different sources, the grouping of different sources to a "virtual database" which looks from the outside world like a single database becomes possible in a very easy manner. Moreover, each of the individual sources grouped together in such a "virtual database" may itself consist of a grouping of different sources.

Therefore, in a tree view as seen by a user and as illustrated in Fig. 2A, in accordance with the scheme of Fig. 6 there may be a further leaf labelled "merging source" as shown in Fig. 2B which indicates a source consisting of a plurality of sources of different kinds merged together through a demultiplex operation performed by the merging driver.

In connection with Fig. 4 there has been described how the merging driver operates for three different sources 430, 440, and 450. The sources on which the merging driver operates may be selected or specified by a user to configure the merging driver. Such a configuration of the merging driver will now be described in somewhat more detail in the following.

Fig. 7 in its upper half shows the steps necessary for configuring a merging driver. At first in step 700 a user has to select the sources which are to be merged. In step 710 then there is created a merging driver based on the selections made by the user. This merging driver is then in step 720 used to access the multiple sources, and in step 730 it obtains a result from the thereby accessed sources.

Fig. 8 schematically shows a merging driver created in accordance with the user's selections for accessing two different sources 825 and 835, those sources being not a part of the merging driver 800. Merging driver 800 contains a receiving module 810 for receiving queries or access instructions from a user interface (or any application program), modules 820 and 830 for distributing the corresponding query to the sources 825 and 835 (not being a part of the merging driver) as specified by the user and for obtaining the results from those sources, and it further contains a module 840 for ordering the results which are retrieved from the different sources and for returning the ordered result to the user interface.

It should be mentioned that such a merging driver as shown in Fig. 8 which has been created according to the user's preferences may be stored and

may be given a certain name, so that it is available at any later stage for simultaneous access to the multiple sources for which it has been configured.

Once a merging driver has been configured and having been labelled by a certain name, it then shows up with its corresponding name in the tree view of Fig. 2B. It can then be used for accessing those database sources for which it has been configured by a single query.

The step of selecting the merged sources is now explained in somewhat more detail in connection with Fig. 9. Fig. 9 shows a window 900 displayed to the user to enable him to configure the merging driver. In the left-hand part 910 all possible sources are listed, and by means of buttons 920 and 930 the user can select or deselect some of them, the selected ones then showing up in the right-hand part 940. By pushing the OK-button 950 the user can complete the selection.

The operation of a so configured merging driver for three different sources is schematically illustrated in Fig. 10. The user inputs a single query into user interface 105 through which the merging driver 110 is accessed. The merging driver then generates multiple queries for the individual sources by distributing the single query to the different sources in step 120, one query for each of the different sources requiring different drivers. Thereby through the generation of multiple queries and the usage of corresponding multiple drivers with an identical API a query result is obtained and ordered in step 130 and is output in step 140.

The merging driver may be used in connection with an option for autocompletion, which means that after having input only a part of the desired query result it is automatically completed based on what is returned from the source. Such an autocompletion procedure, however, requires a lot of RAM, and therefore there may for example be configured one merging driver for those sources only for which the autocompletion makes sense. Another merging driver

which works without autocompletion may be configured for a different combination of database sources.

In case of the above-described "virtual" database which can be queried
5 by a single query through the merging driver being an address book, the user can based on a single query generate serial letters or form letters although the data he needs for generating the serial letters may be distributed over multiple database sources actually requiring different drivers for being accessed.

10 It should further be mentioned that the merging driver may not only be accessed by a user interface, but by any application program which is capable of accessing the API which the merging driver offers to the outside world for accessing it. Such an application program may for example be a word processor, it might even be again a database program, or any other application program for which it makes
15 sense to access the "virtual database" described hereinbefore. It should further be mentioned that the merging driver not only can be used for distributing a single query to multiple sources, but rather that any operation (any access operation) to be performed on multiple sources (such as updating or editing data in the sources) can be carried out based on a single access operation input which then is
20 distributed by the merging driver.

Those skilled in the art will readily recognize that the individual steps and elements of the preferred embodiments of the invention as explained before can be implemented by a software tool or a computer program, and they then represent
25 corresponding computer program code portions which cause the CPU of the computer to execute instructions which finally represent the corresponding steps or elements of the present invention. It will be understood that the steps of Fig. 8 (and other flow charts in this document as well as elements of the invention shown therein) represent actions performed by the CPU of a computer in accordance with
30 computer instructions of a software tool or a computer program.

The computer instructions can also be stored on a computer-readable medium capable of embodying computer instructions, such as a CD-ROM, a floppy-disk, or any transmission link like a connection to the internet, a link to a LAN, etc.

Application code or computer program instructions may be embodied in any form of a computer program product. A computer program product comprises a medium configured to store or transport computer-readable code, or in which computer-readable code may be embedded. Some examples of computer program products are CD-ROM disks, ROM cards, floppy disks, magnetic tapes, computer hard drives, servers on a network, and carrier waves. It may also comprise any signal which does not use carrier waves, such as digital signals transmitted over a network (including the internet) without use of a carrier wave. It may then be in the form of any transmission link suitable to transmit information from one location to another, such as a link through a public switched telephone network, a LAN connection, a WAN connection, or the like. This may lead to an implementation where some or all elements of the invention are implemented on a server, and some or none of the elements of the invention are implemented on a client accessing the server. Through access to the server, the client may then in cooperation with the server implement a computer program embodying an implementation of the present invention or it may use a computer program embodying an implementation of the present invention by accessing the server.

While the present invention has been explained in connection with certain embodiments thereof, other embodiments will be apparent to those skilled in the art from consideration of the specification and practice of the invention disclosed therein. It is intended that the specification and examples be considered as exemplary only, without limiting the spirit and scope of the invention.

1997, 1998, 1999, 2000, 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009, 2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023, 2024, 2025, 2026, 2027, 2028, 2029, 2030, 2031, 2032, 2033, 2034, 2035, 2036, 2037, 2038, 2039, 2040, 2041, 2042, 2043, 2044, 2045, 2046, 2047, 2048, 2049, 2050, 2051, 2052, 2053, 2054, 2055, 2056, 2057, 2058, 2059, 2060, 2061, 2062, 2063, 2064, 2065, 2066, 2067, 2068, 2069, 2070, 2071, 2072, 2073, 2074, 2075, 2076, 2077, 2078, 2079, 2080, 2081, 2082, 2083, 2084, 2085, 2086, 2087, 2088, 2089, 2090, 2091, 2092, 2093, 2094, 2095, 2096, 2097, 2098, 2099, 2100, 2101, 2102, 2103, 2104, 2105, 2106, 2107, 2108, 2109, 2110, 2111, 2112, 2113, 2114, 2115, 2116, 2117, 2118, 2119, 2120, 2121, 2122, 2123, 2124, 2125, 2126, 2127, 2128, 2129, 2130, 2131, 2132, 2133, 2134, 2135, 2136, 2137, 2138, 2139, 2140, 2141, 2142, 2143, 2144, 2145, 2146, 2147, 2148, 2149, 2150, 2151, 2152, 2153, 2154, 2155, 2156, 2157, 2158, 2159, 2160, 2161, 2162, 2163, 2164, 2165, 2166, 2167, 2168, 2169, 2170, 2171, 2172, 2173, 2174, 2175, 2176, 2177, 2178, 2179, 2180, 2181, 2182, 2183, 2184, 2185, 2186, 2187, 2188, 2189, 2190, 2191, 2192, 2193, 2194, 2195, 2196, 2197, 2198, 2199, 2200, 2201, 2202, 2203, 2204, 2205, 2206, 2207, 2208, 2209, 2210, 2211, 2212, 2213, 2214, 2215, 2216, 2217, 2218, 2219, 2220, 2221, 2222, 2223, 2224, 2225, 2226, 2227, 2228, 2229, 2230, 2231, 2232, 2233, 2234, 2235, 2236, 2237, 2238, 2239, 2240, 2241, 2242, 2243, 2244, 2245, 2246, 2247, 2248, 2249, 2250, 2251, 2252, 2253, 2254, 2255, 2256, 2257, 2258, 2259, 2260, 2261, 2262, 2263, 2264, 2265, 2266, 2267, 2268, 2269, 2270, 2271, 2272, 2273, 2274, 2275, 2276, 2277, 2278, 2279, 2280, 2281, 2282, 2283, 2284, 2285, 2286, 2287, 2288, 2289, 2290, 2291, 2292, 2293, 2294, 2295, 2296, 2297, 2298, 2299, 2300, 2301, 2302, 2303, 2304, 2305, 2306, 2307, 2308, 2309, 2310, 2311, 2312, 2313, 2314, 2315, 2316, 2317, 2318, 2319, 2320, 2321, 2322, 2323, 2324, 2325, 2326, 2327, 2328, 2329, 2330, 2331, 2332, 2333, 2334, 2335, 2336, 2337, 2338, 2339, 2340, 2341, 2342, 2343, 2344, 2345, 2346, 2347, 2348, 2349, 2350, 2351, 2352, 2353, 2354, 2355, 2356, 2357, 2358, 2359, 2360, 2361, 2362, 2363, 2364, 2365, 2366, 2367, 2368, 2369, 2370, 2371, 2372, 2373, 2374, 2375, 2376, 2377, 2378, 2379, 2380, 2381, 2382, 2383, 2384, 2385, 2386, 2387, 2388, 2389, 2390, 2391, 2392, 2393, 2394, 2395, 2396, 2397, 2398, 2399, 2400, 2401, 2402, 2403, 2404, 2405, 2406, 2407, 2408, 2409, 2410, 2411, 2412, 2413, 2414, 2415, 2416, 2417, 2418, 2419, 2420, 2421, 2422, 2423, 2424, 2425, 2426, 2427, 2428, 2429, 2430, 2431, 2432, 2433, 2434, 2435, 2436, 2437, 2438, 2439, 2440, 2441, 2442, 2443, 2444, 2445, 2446, 2447, 2448, 2449, 2450, 2451, 2452, 2453, 2454, 2455, 2456, 2457, 2458, 2459, 2460, 2461, 2462, 2463, 2464, 2465, 2466, 2467, 2468, 2469, 2470, 2471, 2472, 2473, 2474, 2475, 2476, 2477, 2478, 2479, 2480, 2481, 2482, 2483, 2484, 2485, 2486, 2487, 2488, 2489, 2490, 2491, 2492, 2493, 2494, 2495, 2496, 2497, 2498, 2499, 2500, 2501, 2502, 2503, 2504, 2505, 2506, 2507, 2508, 2509, 2510, 2511, 2512, 2513, 2514, 2515, 2516, 2517, 2518, 2519, 2520, 2521, 2522, 2523, 2524, 2525, 2526, 2527, 2528, 2529, 2530, 2531, 2532, 2533, 2534, 2535, 2536, 2537, 2538, 2539, 2540, 2541, 2542, 2543, 2544, 2545, 2546, 2547, 2548, 2549, 2550, 2551, 2552, 2553, 2554, 2555, 2556, 2557, 2558, 2559, 2560, 2561, 2562, 2563, 2564, 2565, 2566, 2567, 2568, 2569, 2570, 2571, 2572, 2573, 2574, 2575, 2576, 2577, 2578, 2579, 2580, 2581, 2582, 2583, 2584, 2585, 2586, 2587, 2588, 2589, 2590, 2591, 2592, 2593, 2594, 2595, 2596, 2597, 2598, 2599, 2600, 2601, 2602, 2603, 2604, 2605, 2606, 2607, 2608, 2609, 2610, 2611, 2612, 2613, 2614, 2615, 2616, 2617, 2618, 2619, 2620, 2621, 2622, 2623, 2624, 2625, 2626, 2627, 2628, 2629, 2630, 2631, 2632, 2633, 2634, 2635, 2636, 2637, 2638, 2639, 2640, 2641, 2642, 2643, 2644, 2645, 2646, 2647, 2648, 2649, 2650, 2651, 2652, 2653, 2654, 2655, 2656, 2657, 2658, 2659, 2660, 2661, 2662, 2663, 2664, 2665, 2666, 2667, 2668, 2669, 2670, 2671, 2672, 2673, 2674, 2675, 2676, 2677, 2678, 26

1. The first step in the process of identifying a problem is to define the problem. This involves identifying the symptoms of the problem and determining the scope of the problem. Once the problem has been defined, the next step is to identify the causes of the problem. This involves identifying the factors that are contributing to the problem and determining the underlying causes. Once the causes have been identified, the next step is to develop a plan of action. This involves identifying the steps that need to be taken to solve the problem and determining the resources that will be needed to implement the plan. Finally, the last step in the process is to implement the plan and monitor the results. This involves putting the plan into action and tracking the progress of the solution to ensure that the problem is solved.

1. The first step in the process is to identify the problem or issue that needs to be addressed. This involves gathering information and understanding the context of the situation.

1. The results of the study are consistent with the findings of other studies that suggest that the use of a structured approach to problem solving can lead to improved outcomes in the management of complex problems.

This Page Blank (uspto)

ALL INFORMATION CONTAINED HEREIN IS UNCLASSIFIED
DATE 08-11-2010 BY 60322 UCBAW

1. The first step in the process of identifying a problem is to define the problem. This involves identifying the symptoms of the problem and determining the scope of the problem. Once the problem has been defined, the next step is to identify the causes of the problem. This involves identifying the factors that are contributing to the problem and determining the underlying causes of the problem. Once the causes of the problem have been identified, the next step is to develop a plan to address the problem. This involves identifying the actions that need to be taken to address the problem and determining the resources that will be needed to implement the plan. Once a plan has been developed, the next step is to implement the plan. This involves taking the actions that have been identified in the plan and putting them into practice. Finally, the last step in the process is to evaluate the results of the plan. This involves determining whether the plan has been successful in addressing the problem and identifying any areas for improvement.

1989年12月25日
 1990年1月1日

1990年12月15日 星期一 晴

14. Dez. 1999

WHAT IS CLAIMED IS:

1. A method for enabling access of multiple sources (430, 440, 450) of a database (200) by a single access, said multiple sources requiring to be accessed
5 by different drivers (425, 435, 445), said method comprising:
 providing a common API (420) for individually accessing said multiple sources through said corresponding drivers; and
 providing a merging driver (410) for accessing a plurality of said sources by a single access through said common API respectively accessing said multiple
10 sources through said corresponding drivers.
2. The method of claim 1, wherein said merging driver (410) distributes a single access instruction to a plurality of said different drivers (425, 435, 445).
15
3. The method of claim 1, further comprising:
 obtaining a selection of the multiple sources to be accessed by a single access from a user to configure said merging driver.
- 20 4. The method of claim 1, wherein said single access is a search or a query in said database.
5. The method of claim 1, further comprising:
 generating multiple queries by said merging driver, each of said query
25 accessing one of said plurality of sources to be accessed by said merging driver.
6. The method of claim 1, further comprising:
 obtaining an ordered result of said access.
- 30 7. The method of claim 6, wherein obtaining the ordered result comprises:

if more than one results are offered, obtaining the results according to their priority based on an ordering criterion.

8. The method of claim 1, further comprising:
5 performing an autocompletion for the outputs obtained through said single access.

9. The method of claim 1, wherein:
said merging driver (550) forms one of the sources which can be
10 accessed through said common API (510);

10. A software tool for enabling access of multiple sources (430, 440, 450) of a database (200) by a single access, said multiple sources requiring to be accessed by different drivers (425, 435, 445), said software tool comprising:
15 a portion configured for providing a common API (420) for individually accessing said multiple sources through said corresponding drivers; and
a portion configured for providing a merging driver (410) for accessing a plurality of said sources by a single access through said common API respectively accessing said multiple sources through said corresponding drivers.
20

11. The software tool of claim 10, wherein said merging driver (410) distributes a single access instruction to a plurality of said different drivers (425, 435, 445).

25 12. The software tool of claim 10, further comprising:
a portion configured for obtaining a selection of the multiple sources to be accessed by a single access from a user to configure said merging driver.

13. The software tool of claim 10, wherein said single access is a
30 search or a query in said database.

14. The software tool of claim 10, further comprising:

a portion configured for generating multiple queries by said merging driver, each of said query accessing one of said plurality of sources to be accessed by said merging driver.

5 15. The software tool of claim 10, further comprising:
a portion configured for obtaining an ordered result of said access:

16. The software tool of claim 15; wherein obtaining the ordered result comprises:
10 if more than one results are offered, obtaining the results according to their priority based on an ordering criterion.

17. The software tool of claim 10, further comprising:
a portion configured for performing an autocompletion for the outputs
15 obtained through said single access:

18. The software tool of claim 10, wherein:
said merging driver forms one of the sources which can be accessed
through said common API.

19. A computer program for enabling access of multiple sources (430, 440, 450) of a database (200) by a single access, said multiple sources requiring to be accessed by different drivers (425, 435, 445), said computer program comprising:

25 a computer program code portion for providing a common API (420) for individually accessing said multiple sources through said corresponding drivers; and

a computer program code portion providing a merging driver (410) for accessing a plurality of said sources by a single access through said common API
30 respectively accessing said multiple sources through said corresponding drivers.

20. The computer program of claim 19, wherein said merging driver (410) distributes a single access instruction to a plurality of said different drivers (425, 435, 445).

5 21. The computer program of claim 19, further comprising:
a computer program code for obtaining a selection of the multiple sources to be accessed by a single access from a user to configure said merging driver.

10 22. A computer program product comprising computer program code for enabling access of multiple sources (430, 440, 450) of a database (200) by a single access, said multiple sources requiring to be accessed by different drivers (425, 435, 445), said computer program code comprising:

15 a computer program code portion for providing a common API (420) for individually accessing said multiple sources through said corresponding drivers; and

20 a computer program code portion providing a merging driver (410) for accessing a plurality of said sources by a single access through said common API respectively accessing said multiple sources through said corresponding drivers.

23. The computer program product of claim 22, wherein said merging driver (410) distributes a single access instruction to a plurality of said different drivers (425, 435, 445).

25 24. The computer program product of claim 22, further comprising:
a computer program code portion for obtaining a selection of the multiple sources to be accessed by a single access from a user to configure said merging driver.

30 25. The computer program product of claim 22, wherein at least a part of said computer program code is implemented on a server accessed by a client on which the remaining part of said computer program code is implemented.

26. A computer programmed such that it is capable of carrying out a method for enabling access of multiple sources (430, 440, 450) of a database (200) by a single access, said multiple sources requiring to be accessed by different drivers (425, 435, 445), said method comprising:

providing a common API (420) for individually accessing said multiple sources through said corresponding drivers; and

providing a merging driver (410) for accessing a plurality of said sources by a single access through said common API respectively accessing said multiple sources through said corresponding drivers.

The method of claim 26, wherein the merging driver (410) is configured to receive a single access request from a user and to route the request to the appropriate driver (425, 435, 445) for processing.

The method of claim 26, wherein the merging driver (410) is configured to aggregate data from the multiple sources into a single result set.

The method of claim 26, wherein the merging driver (410) is configured to perform a join operation on the data from the multiple sources.

The method of claim 26, wherein the merging driver (410) is configured to perform a union operation on the data from the multiple sources.

The method of claim 26, wherein the merging driver (410) is configured to perform a select operation on the data from the multiple sources.

The method of claim 26, wherein the merging driver (410) is configured to perform a delete operation on the data from the multiple sources.

The method of claim 26, wherein the merging driver (410) is configured to perform an insert operation on the data from the multiple sources.

The method of claim 26, wherein the merging driver (410) is configured to perform a rename operation on the data from the multiple sources.

The method of claim 26, wherein the merging driver (410) is configured to perform a truncate operation on the data from the multiple sources.

The method of claim 26, wherein the merging driver (410) is configured to perform a drop operation on the data from the multiple sources.

The method of claim 26, wherein the merging driver (410) is configured to perform a create operation on the data from the multiple sources.

The method of claim 26, wherein the merging driver (410) is configured to perform a alter operation on the data from the multiple sources.

The method of claim 26, wherein the merging driver (410) is configured to perform a grant operation on the data from the multiple sources.

The method of claim 26, wherein the merging driver (410) is configured to perform a revoke operation on the data from the multiple sources.

The method of claim 26, wherein the merging driver (410) is configured to perform a flush operation on the data from the multiple sources.

The method of claim 26, wherein the merging driver (410) is configured to perform a set operation on the data from the multiple sources.

The method of claim 26, wherein the merging driver (410) is configured to perform a show operation on the data from the multiple sources.

This Page Blank (uspto)

EPO - Munich
61

1/10

14. Dez. 1999

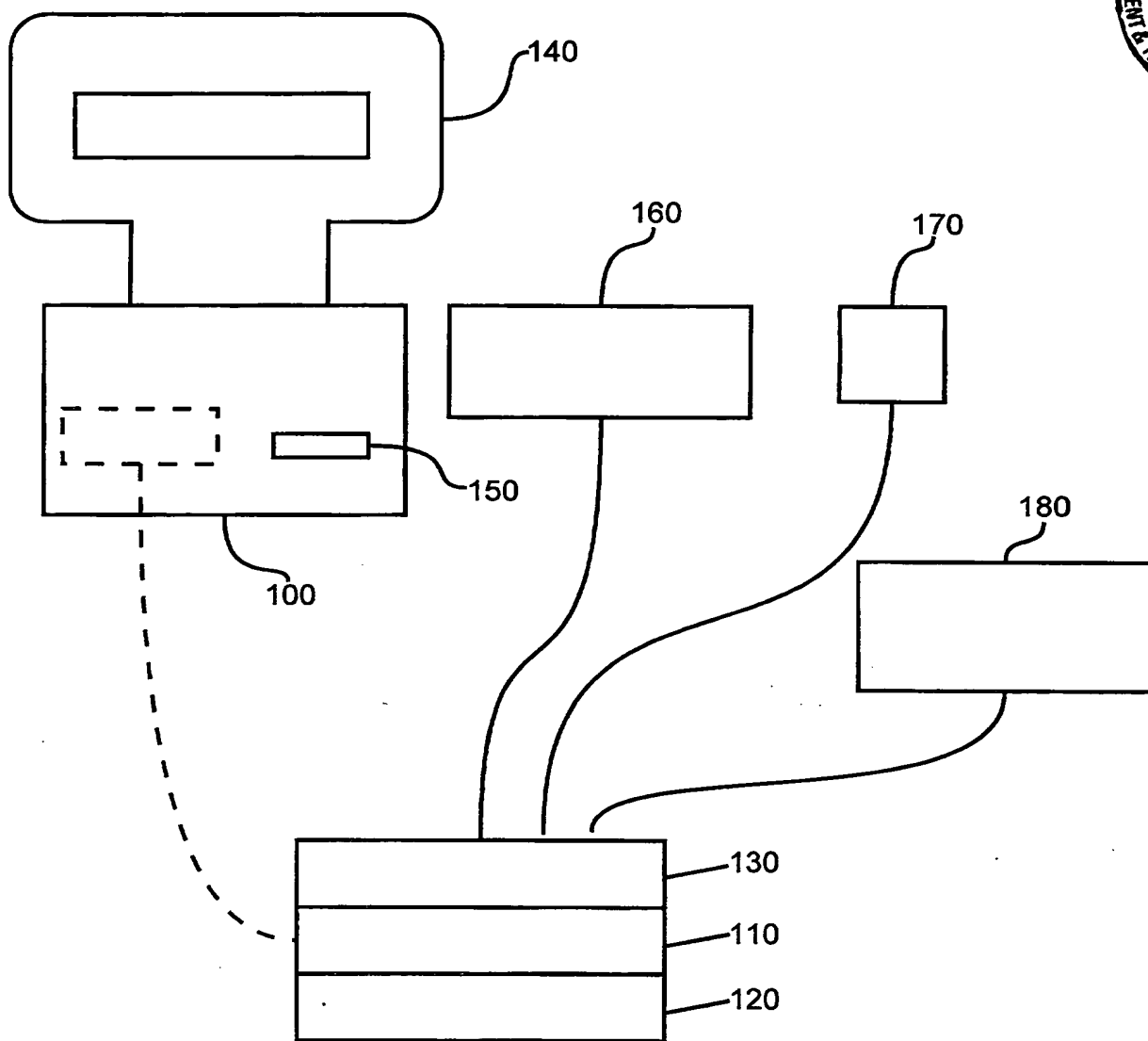
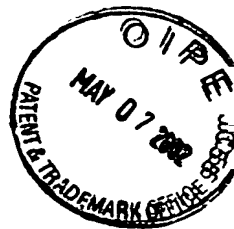


Fig. 1



2/10

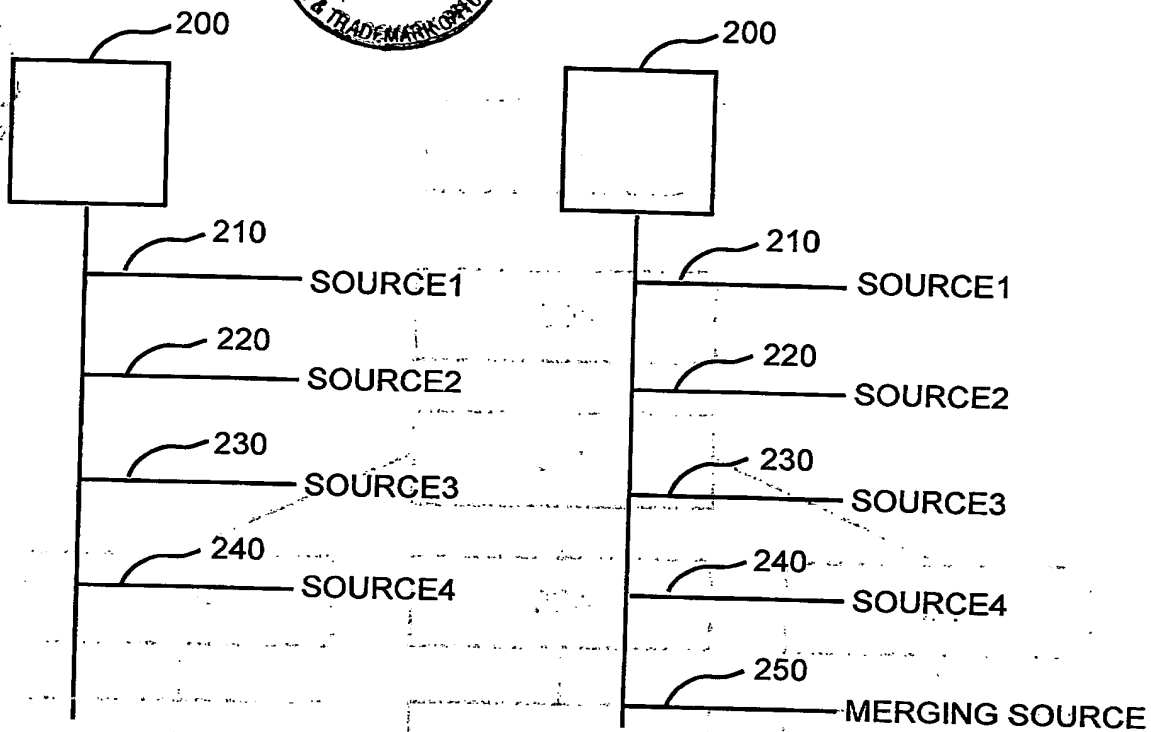


Fig. 2A

Fig. 2B

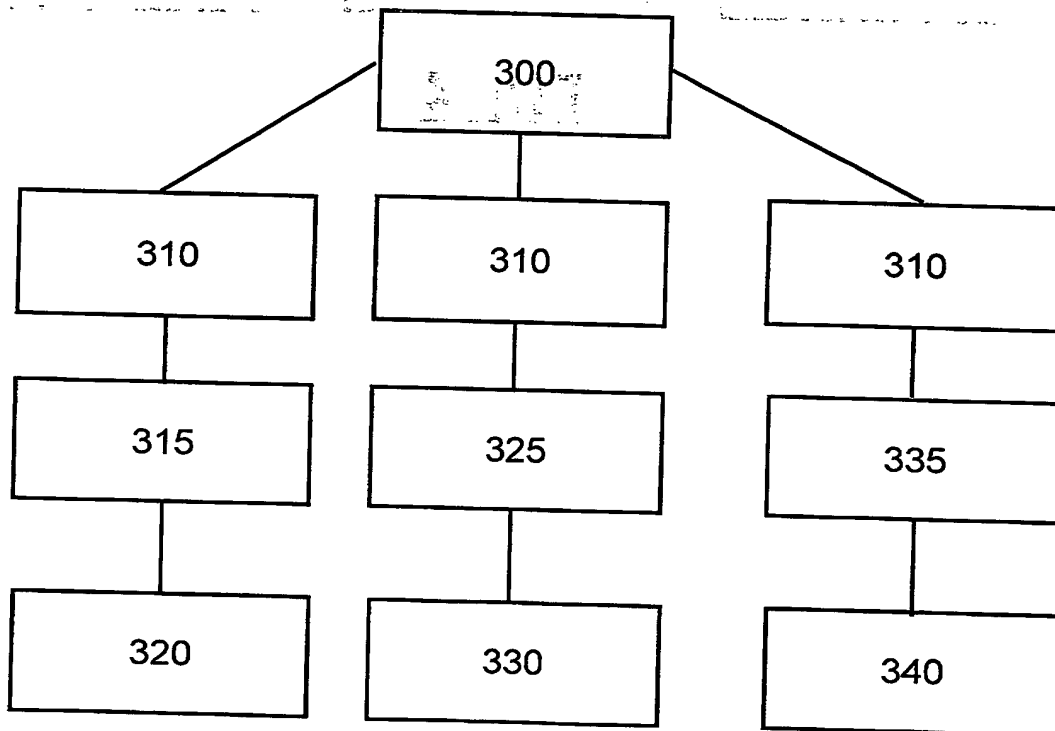
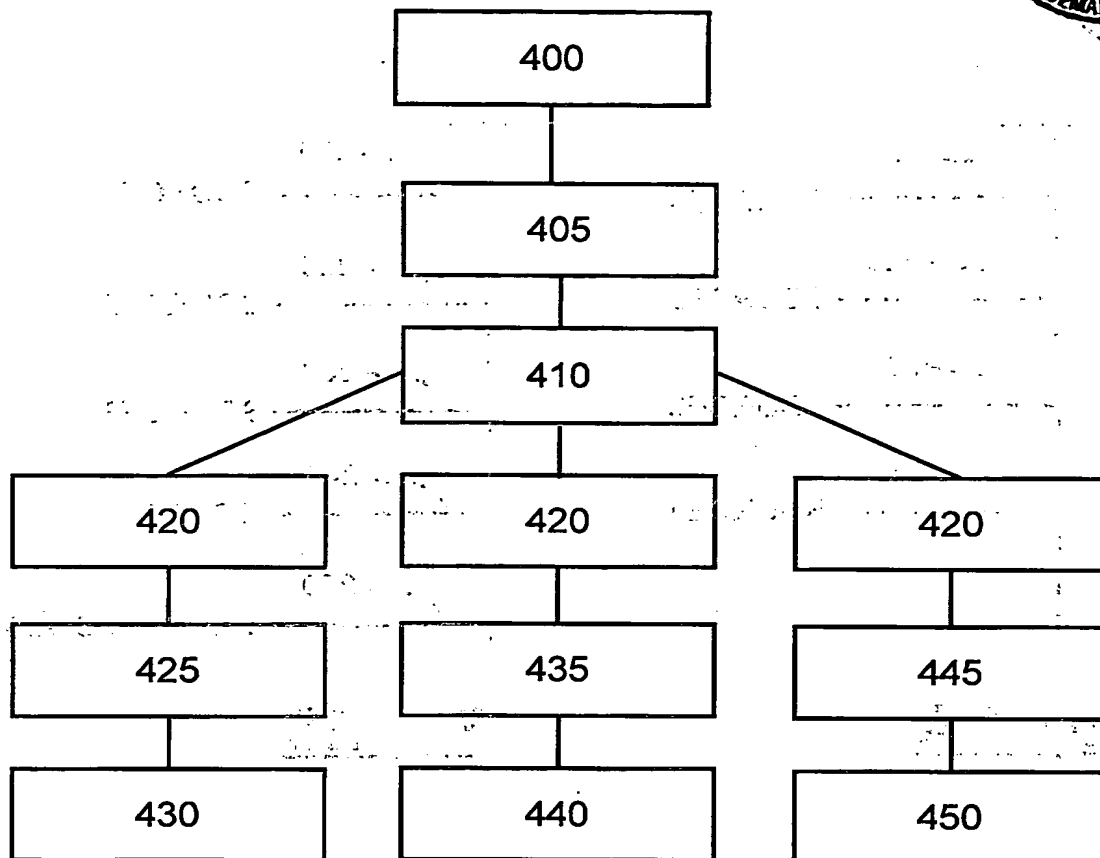
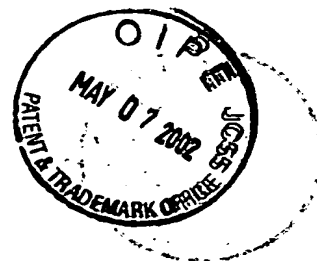


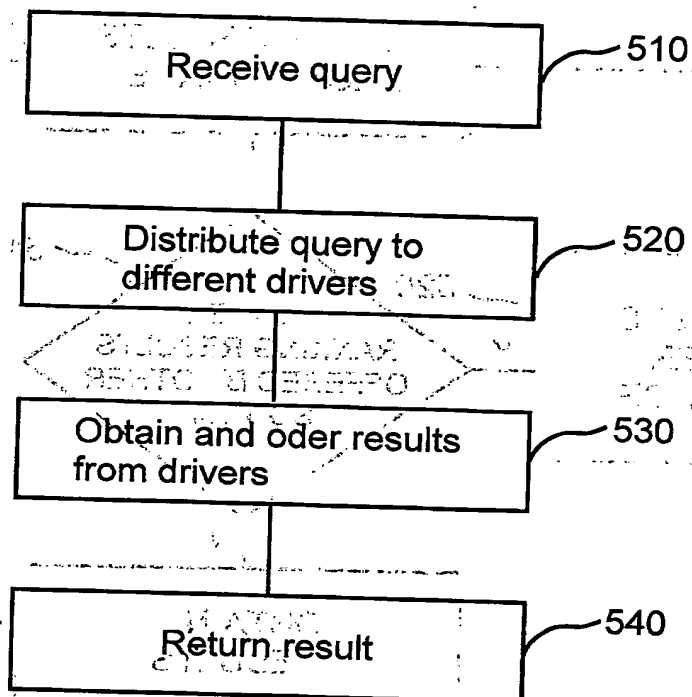
Fig. 3

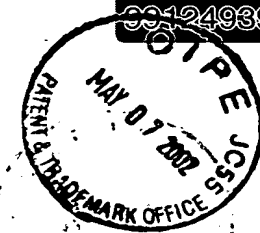
3/10

**Fig. 4**

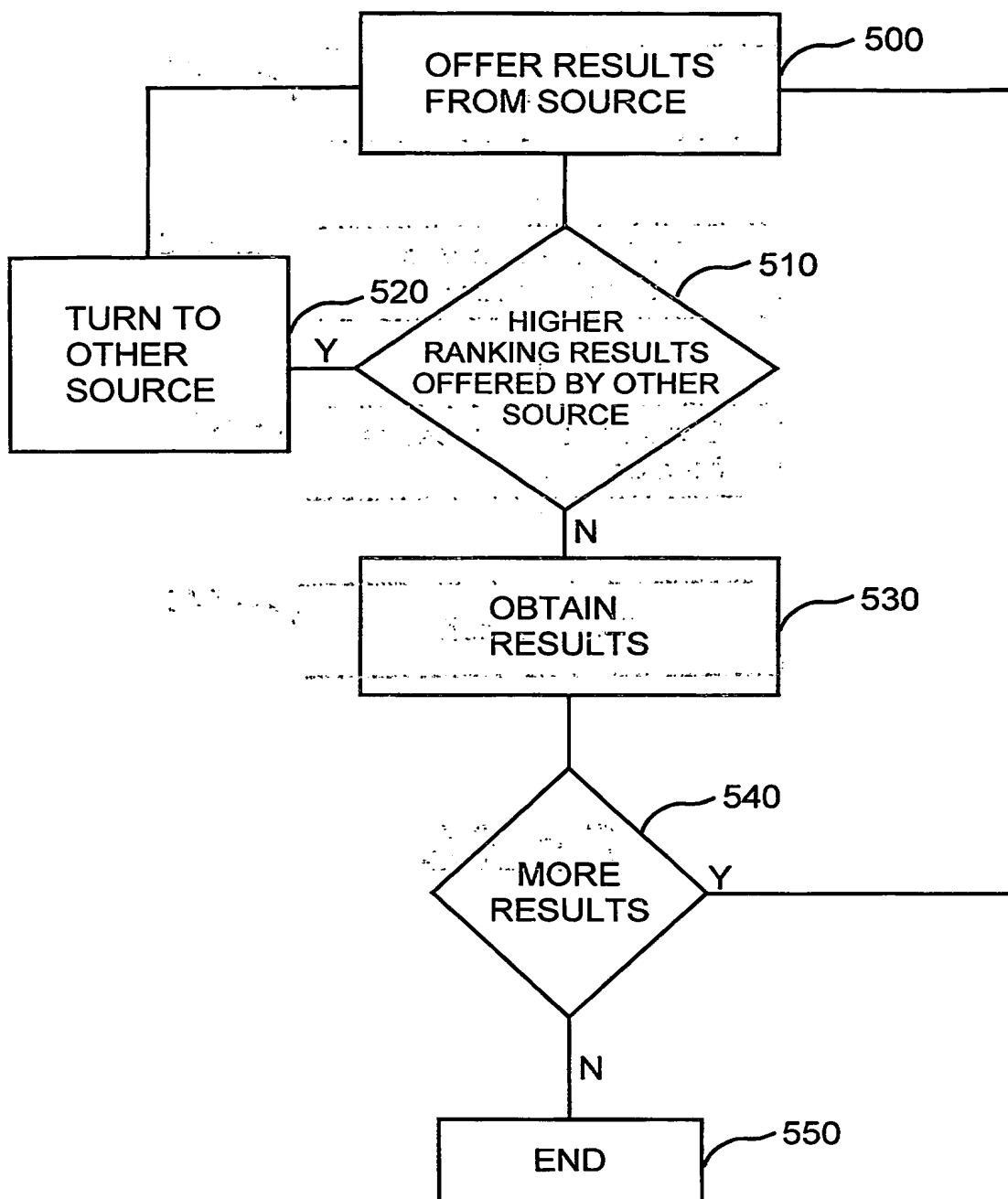


4/10

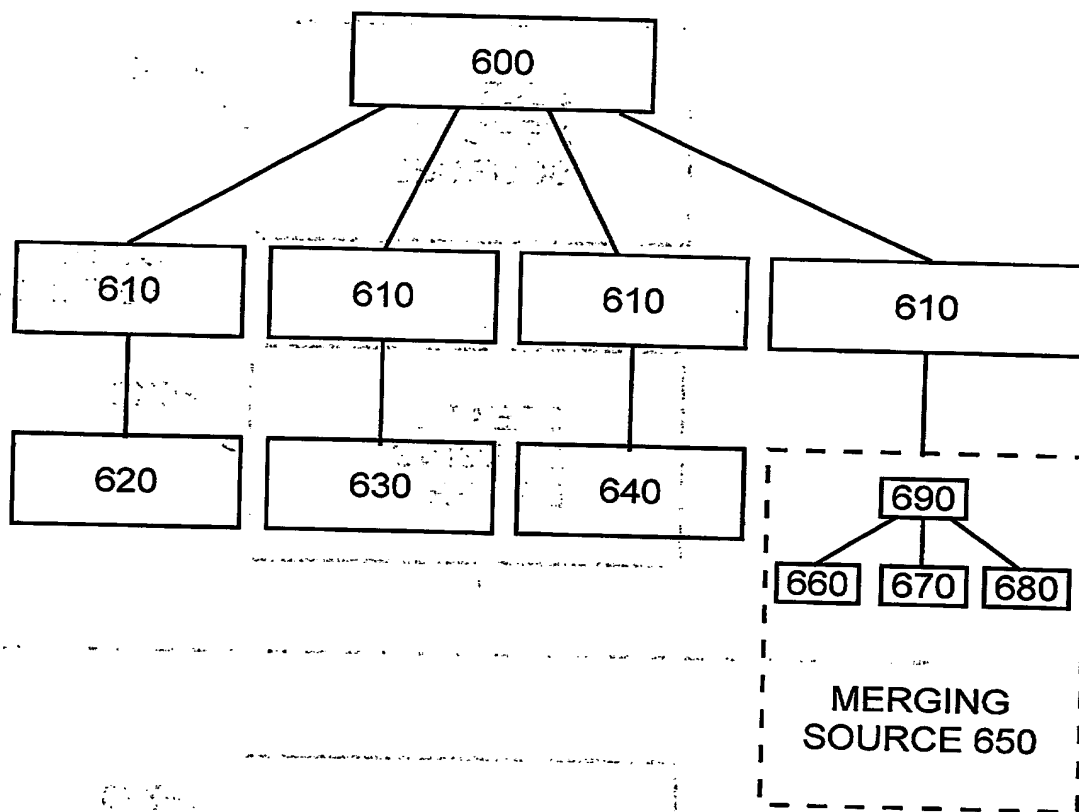
**Fig. 5A**

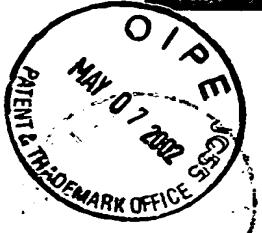


5/10

**Fig. 5B**

6/10

**Fig. 6**



7/10

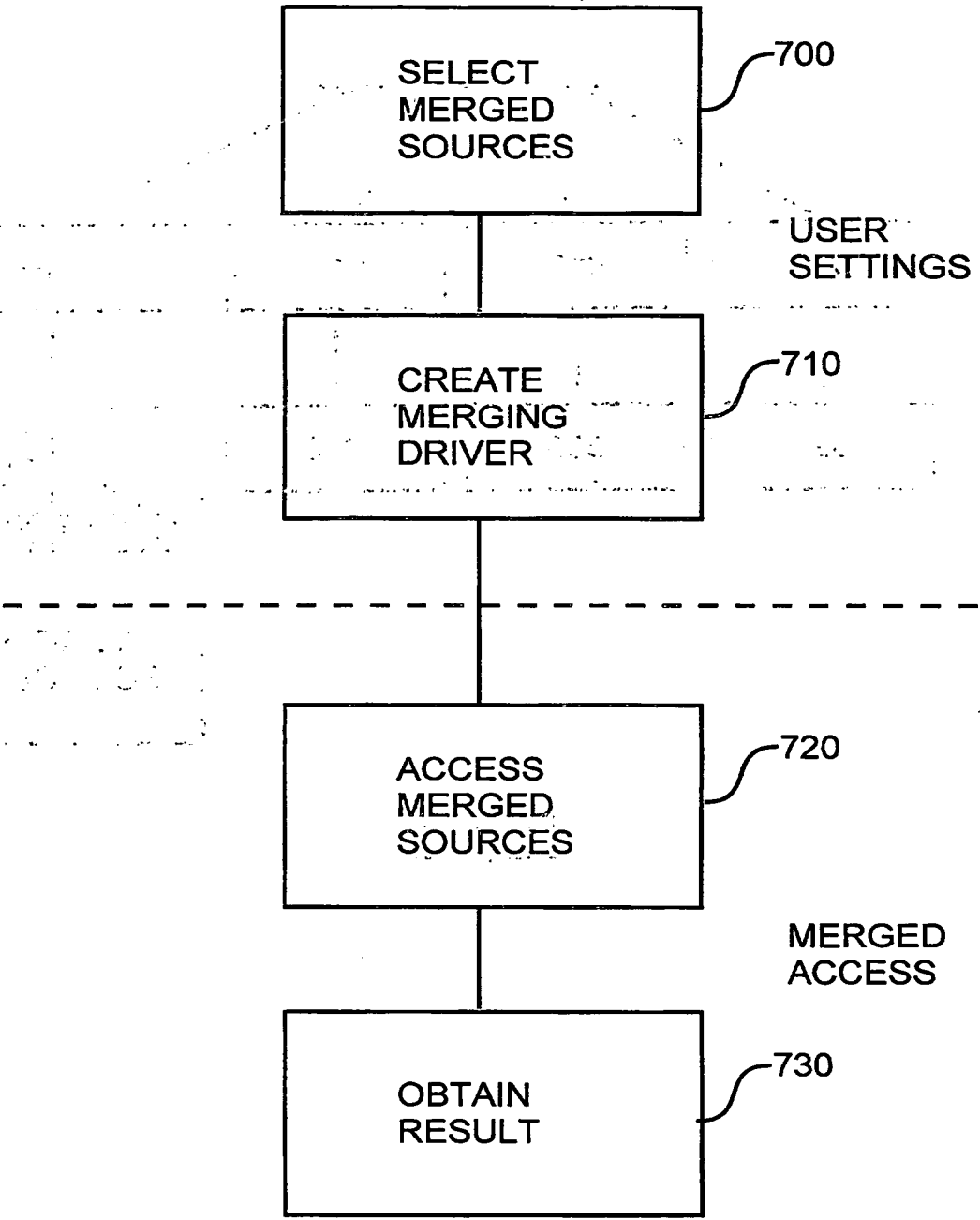
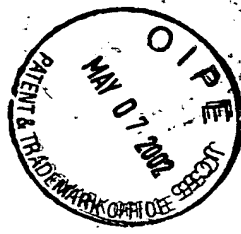


Fig. 7



8/10

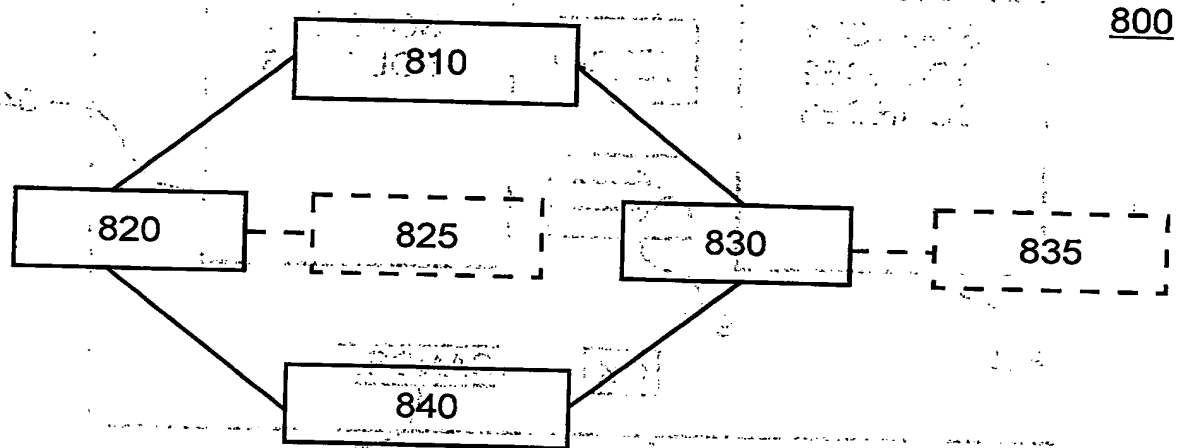


Fig. 8

9/10

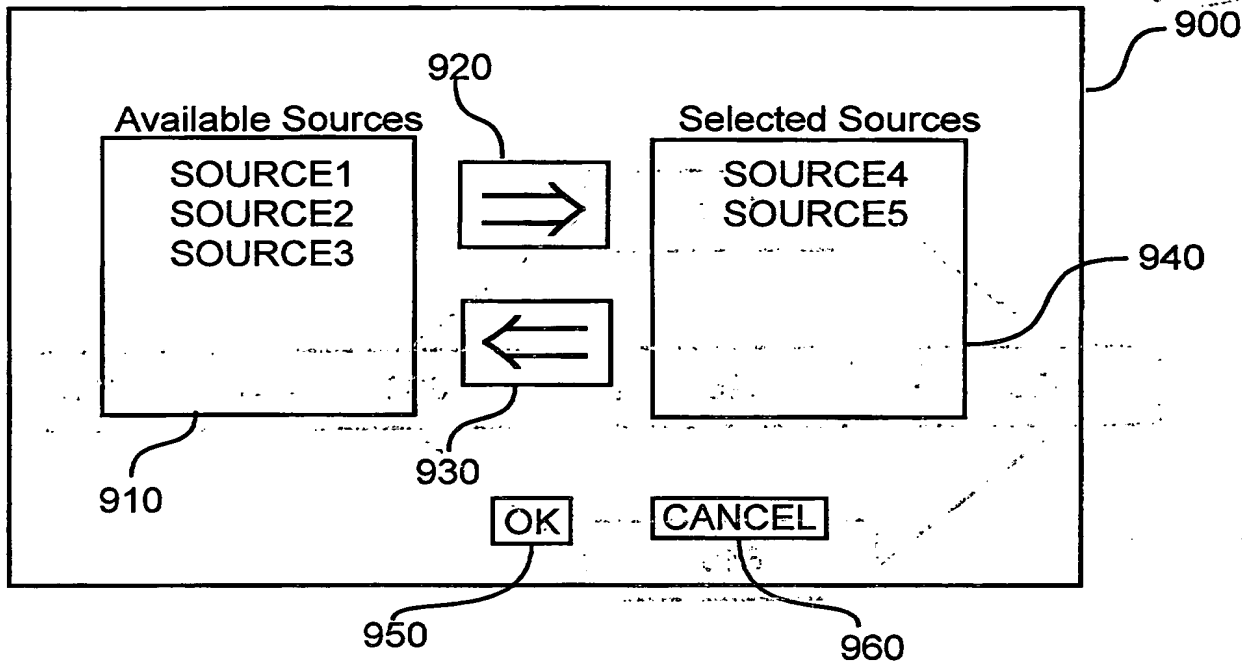
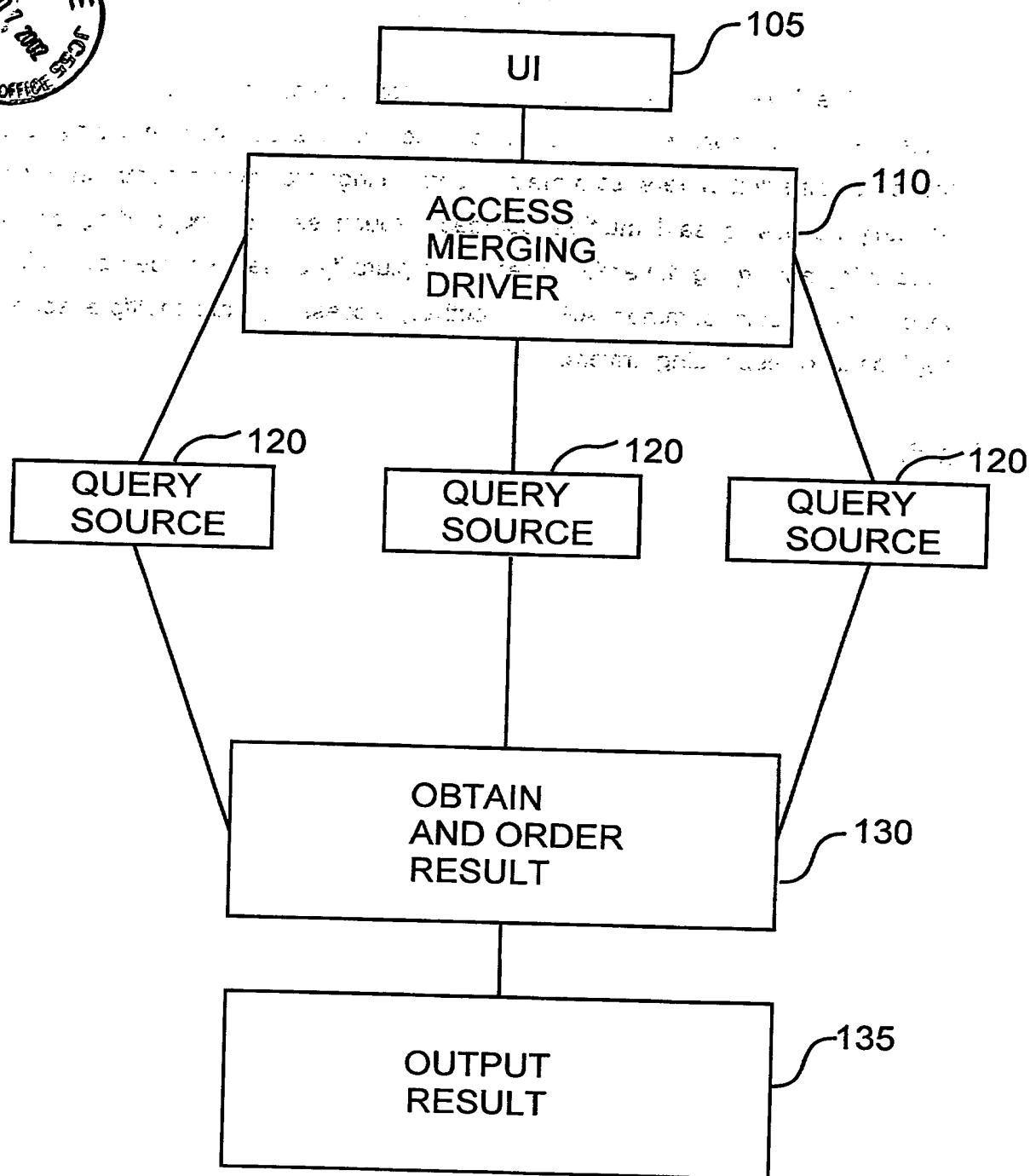


Fig. 9

10/10

**Fig. 10**

14. Dez. 1999

ABSTRACT

The invention relates to a method for accessing by a single access multiple database sources of a database, said multiple sources requiring to be
5 accessed by different drivers, said method comprising: providing a common API for individually accessing said multiple sources through said corresponding drivers; and providing a merging driver for accessing a plurality of said sources by a single access through said common API respectively accessing said multiple sources through said corresponding drivers.

10

(Fig. 6)

FIG. 6 is a block diagram illustrating a method for accessing multiple database sources. It shows a central block labeled 'Merging Driver' which is connected to a 'Common API' block. The 'Common API' block is then connected to multiple 'Database Sources' blocks, each of which is associated with a specific 'Driver' block. The diagram illustrates how a single access through the common API is used to access multiple sources via their respective drivers.

FIG. 7 is a block diagram illustrating a method for accessing multiple database sources. It shows a central block labeled 'Merging Driver' which is connected to a 'Common API' block. The 'Common API' block is then connected to multiple 'Database Sources' blocks, each of which is associated with a specific 'Driver' block. The diagram illustrates how a single access through the common API is used to access multiple sources via their respective drivers.

FIG. 8 is a block diagram illustrating a method for accessing multiple database sources. It shows a central block labeled 'Merging Driver' which is connected to a 'Common API' block. The 'Common API' block is then connected to multiple 'Database Sources' blocks, each of which is associated with a specific 'Driver' block. The diagram illustrates how a single access through the common API is used to access multiple sources via their respective drivers.



This Page Blank (uspto)